Vol.2011-CSEC-55 No.7
2011/12/5

パブサブシステムにおける安全な統計情報の計算

南　和宏[†1]　アダム　リー[†3]
ニキータ　ボリソブ[†2]　マリアン　ウインズレット[†2]

パブサブシステムの内部ノードがパブリッシャーからの入力データに基づきサブスクライバーの必要とする統計情報を計算させることにより、システムのネットワークの負荷を軽減することが可能である。しかし大規模なパブシステムのノードは単一の管理ドメインに属するとは限らず、情報の機密性、正確性に関して信頼することはできない。本発表では、信頼できないノードが複数の入力数値データの合計を計算する状況において、1) パブリッシャーのデータの機密性、2) サブスクライバーの受け取る合計値の正確性を保証する分散プロトコルを提案する。このプロトコルは、homomorphic な Message Authentication Code (MAC) を用い、サブスクライバーに対し、個々の入力データの電子署名を入手することなく合計値の正確性を検証することを可能にした.

Secure Aggregation in a Publish-subscribe system

Kazuhiro Minami,[†1] Adam J. Lee,[†3] Nikita Borisov[†3]
and Marianne Winslett

A publish-subscribe system is an information dissemination infrastructure that supports many-to-many communications among publishers and subscribers. In many publish-subscribe systems, in-network aggregation of input data is considered to be an important service that reduces the bandwidth requirements of the system significantly. In this paper, we present a scheme for securing the aggregation of inputs to such a publish-subscribe system. Our scheme—which focuses on the additive aggregate function *sum*—preserves the confidentiality and integrity of aggregated data in the presence of untrusted routing nodes. Our scheme allows a group of publishers to publish aggregate data to authorized subscribers without revealing their individual private inputs to either the routing nodes or the subscribers. In addition, our scheme allows subscribers to verify that routing nodes perform the aggregation operation correctly. We use a message authentication code (MAC) scheme based on the discrete logarithm property to allow subscribers to verify the correctness of aggregated data without receiving the digitally-signed raw data used as input to the aggregation.

1. はじめに

パブサブシステム[1),2),9),12)] は複数の情報提供者（パブリッシャー）と情報の受け手（サブスクライバー）の間を広域のネットワークを介して取り持ち、情報の流通拡散を行うミドルウエアである．パブリッシャーは流通させたい情報をシステムに向けて発信し、サブスクライバーは情報を受けるために自分の興味をシステムに登録する．パブリッシャーから発信されたデータは途中、パブサブシステムが構成するオーバーレイネットワーク上のルーティングノードでパブリッシャーに向けて転送されていく．パブサブシステムは様々なアプリケーションから発信される膨大なデータに対応するために、効率の良い転送ルートを確立し、もし同位置ルート上の重複するデータを取り除く機能をもつ．

本論文では、オーバーレイネットワークのルーティングノードが途中で統計データの計算を行うインネットワークアグリゲーション (In-network aggregation) の問題を検討する．インネットワークアグリゲーションは、スマートグリッド[13)] やインテリジェントビルシステム[3)] のような広域の物理システムの状態をモニターするアプリケーションに対して有用である．物理システムのモニターリングにパブサブシステムを適用する場合、様々な物理量を計測センサーがパブリッシャーの役割を担い、モニターされる情報に基づき物理システムを制御するアプリケーションがパブリッシャーの役割を持つ．場合によっては、広域に設置された多数のセンサーが非常に高い頻度でデータを発信し続けることがある．例えば、電力送電システムで使用される電力波形測定装置 (Phasor measurement unit)[10)] は通常、一秒間にデータを３０回発信する．多くのモニタリングアプリケーションが必要とするのは個々のセンサーデータではなく、ある程度大局的な統計データであることを考えると、パブサブシステムがインネットワークアグリゲーションの機能を提供することで、ネットワークの負荷を軽減することは重要になる．

安全なインネットワークアグリゲーションをパブサブシステム上で実現するためには２つのセキュリティの課題に取り組む必要がある．一つ目は、パブリッシャーから発信される情報の機密性の問題であり、必ずしも信頼できない途中のルーティングノードや末端のサブスクラバーから機密情報を守る必要がある．例えば、電力需

†1 国立情報学研究所
National Institute of Informatics
†2 イリノイ大学
University of Illinois
†3 ピッツバーグ大学
University of Pittsburgh

要をリアルタイムにモニターするスマートグリッドのシステムにおいて、個々の家庭の電力消費の詳細は電力会社に知られるべきではない．なぜなら電力の使用状況からある程度、その家庭での活動の様子が類推できてしまうからである[5]．一般に、広域をカバーするパブサブシステムのルーティングノードは異なる管理ドメインに属することになり、各パブリッシャーはそれらルーティングノードを信頼しないと仮定するのが妥当である．よって本論文では、ルーティングノードがパブリッシャーから発信されるデータを読むことなく統計データの計算を行うインネットワークアグリゲーションのプロトコルを開発することを第一の目的とした．

２つ目は、統計情報を受け取るサブスクラバーに対し、いかに情報の正確性、正当性を保証するかという問題である．データの正確生は制御システムのためのモニターアプリケーションにとって非常に重要である．もしモニタリングしている情報が正しくなければ、物理システムに障害を生じるような誤った制御信号を発信する危険性が生じる．しかし途中のルーティングノードが信頼できない場合、統計情報の正確性を保証するのは困難である．なぜなら攻撃者に管理されたルーティングノードが誤った計算結果を報告する可能性があるからである．一番確実なのは、サブスクライバーが個々のパブリッシャーのデータを入手して、統計データを再計算する方法だが、それでは元々のインネットワークアグリゲーションのネットワーク負荷を軽減するという目的が損なわれてしまう．したがって、サブスクライバーはパブリッシャーの元データを参照することなく統計データの正確生を検証する方法が必要になる．

本論文では、安全なインネットワークアグリゲーションの実現に向けた最初の第一歩として元データの合計 (sum) を算出する安全なアグリゲーションのプロトコルを検討した。合計は単純な関数であるが、他の幾つかの有用な関数（例えば、平均、カウント、標準偏差）に変換することが可能である．我々が開発したプロトコルでは、複数のパブリッシャーのデータから計算された統計データが権限をもつサブスクライバーにのみ配信され、個々のデータがルーティングノード、サブスクライバー等に漏洩することがない．また離散対数の準同型の特性を持つメッセージ認証コードを用いることで、サブスクライバーが統計データの正確性を検証できるようにした．我々の提案する方式では、各パブリッシャーがデータを発信する際に、電子署名を付ける必要がない．

本論文の構成は下記の通りである．第２章では、パブサブシステムの概要と本論文における攻撃モデルを叙述する．第３章では、アグリゲーションのプロトコルを解説する。第４章では関連する研究を概観し、第５章で結論を述べる．

2. システムモデル

2.1 システムの概要

本論文で想定するパブサブシステムは、複数のルーティングノードと信頼できるセキュリティマネージャー（security manager) のノードから構成される．パブリッシャーとサブスクライバーはシステム外に存在し、パブサブシステムのサービスを利用するアプリケーションという位置付けになる．各パブリッシャー、サブスクライバー、そしてルーティングノードはある主体 (principal) によって管理されており、全ての主体は集合 $\mathcal{P}$ に属する．セキュリティマネージャーは全てのパブリッシャー、サブスクライバーから信頼されており、データの機密性に関するセキュリティのポリシーを管理する．各主体 p_i は公開鍵、秘密鍵のペア (K_i, K_i^{-1}) を所持し、Public Key Infrastructure (PKI) のサービスを利用して、他の主体の公開鍵を入手することができる．また各パブリッシャーが発信するデータは全てあるデータの集合 $\mathcal{V}$ に属すると仮定する．

図 1 はパブリッシャー、サブスクライバー、そしてシステムのコンポーネント間のメッセージのやり取りを示す．最初に各パブリッシャーは自身のセキュリティのポリシーをセキュリティマネージャーに通知する．その結果、セキュリティマネージャーは全てのパブリッシャーの機密データに関するセキュリティポリシーを把握する．サブスクライバーがあるデータを受け取りたいときは、事前に購読（サブスクリプション）のリクエストをセキュリティマネージャーに送る必要がある．購読のリクエストは、（主体名、データ名）の複数ペアで記述される．例えば、もし主体 p_0 のデータ v_0 と主体 p_1 のデータ v_1 の合計を受け取りたければ、購読リクエストは $\{(p_0, v_0), (p_1, v_1)\}$ と記述される．本論文のパブサブシステムでは、サブスクライバーがどのパブリッシャーからデータを受け取りたいか明示的に指定するので通常のパブサブシステムよりもパブリッシャーとサブスクライバーの結びつきが強くなる．しかし物理システムを制御するモニターアプリケーションでは、データの信頼性を向上されるために特定のパブリッシャーのみからデータを受けることが重要になる．

セキュリティマネージャーは購読リクエストを受け取ると、リクエストを送信してきたサブスクライバーが各パブリッシャーのセキュリティポリシーを満足していることを確認する．もし条件が満たされているなら、セキュリティマネージャーはパブリッシャーからのデータをサブスクライバーにルーティングするための経路を計算する．途中で統計データの計算を行うので、ルーティングの経路はツリー型の階層構造を形成する．どのように経路が計算されるかは本論文の対象外である．いったん、各パブリッシャーがツリー型経路の末端のルーティングノードにデータを発信

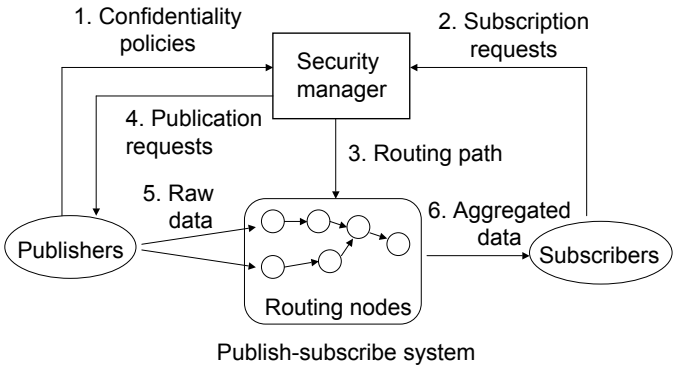

図 1 システムモデル．各矢印はデータの移動を示す．

し始めると、各ルーティングノードは受け取ったデータの合計を計算し、その結果を次のルーティングノードに渡す．最終的には、サブスクライバーに一番近いルートのノードが全体の合計値をサブスクライバーに渡す．

各パブリッシャー p_i は他のパブリッシャーと同期しながら定期的にデータ v_i を発信する．言い換えると全てのパブリッシャーが同期しながら統一時刻 t にデータを発信する．パブリッシャー p_i が時刻 t に発信したデータは、$v_i(t)$ と表現することにする．各ルーターが受け取った複数データの合計値を計算する場合、同一のタイムスタンプのデータのみを対象とする．もし n 個のパブリッシャー $p_1, \ldots, p_n$ がデータ $v_1, \ldots, v_n$ を発信するとするとそれらの合計値を購読するサブスクライバーはそれぞれの時刻 t ごとに、合計値 $\sum_{i=1}^{n} v_i(t)$ を受け取る．

2.2 機密ポリシー

各パブリッシャー p_i は自分のデータの公開を制限する機密ポリシーを定義する．パブリッシャー p_i は各データ v_i に対し、アクセスコントロールリスト $acl_i(v_i)$ を定義する．もし $acl_i(v_i) = \{p_j\}$ であれば、サブスクライバー p_j のみが p_i のデータ v_i を受け取ることができる．

各パブリッシャーは複数のデータの合計値に対して機密ポリシーを定義することができる．この場合、あるデータが別のパブリッシャーから発信されるデータでも構わない．この場合、アクセスコントロールリストは、複数の（主体、データ名）のペアに対して定義される．例えば、パブリッシャー p_i はアクセスコントロールリスト $acl_i((p_i, v_i), (p_j, v_j))$ を定義し、自身のデータ v_i と別の主体 p_j のデータ v_j の合計値に機密ポリシーを定義できる．もしサブスクライバー p_k が前述の合計値に購読リクエストを出した場合、p_k は２つの主体 p_i と p_j の両方の機密ポリシーを満足する必要がある．つまり、下記の２つの条件を満たす必要がある．

$$p_k \in acl_i((p_i, v_i), (p_j, v_j))$$
$$p_k \in acl_j((p_i, v_i), (p_j, v_j))$$

２つ以上のデータの合計値に対しても同様にして機密ポリシーを定義することができる．

2.3 攻撃モデル

パブリッシャーそしてサブスクライバーそれぞれの観点からの攻撃モデルを述べる．パブリッシャーに対しては２種類の攻撃者が存在する．一つは合計値を権限なしに知ろうとするルーティングノードである．複数のノードが共謀する場合も考えられる．２つめは、パブリッシャーの個別のデータを知ろうとする攻撃者である．この攻撃者は、権限のないサブスクライバー、ルーティングノード、さらにはそれらが共謀する場合も考えられる．共謀する複数の主体は互いに自由に情報を交換できる．

攻撃者は、メッセージのやりとりを観察することでルーティングの経路を知ることができる．しかし全ての２者間のメッセージは共有鍵で暗号化されており、攻撃者に属しないノード内のデータを参照することはできない．後者のパブリッシャーの個別のデータを知ろうとする攻撃者を考慮する場合、その攻撃者に属するサブスクライバーとルーティングノードの数は最大 m 個と仮定する．

サブスクライバーの観点からは、攻撃者は誤った合計値を提供するルーティングノードである．この攻撃者は任意の数のルーティングノードを含むことができる．この攻撃者にはパブリッシャーは含まれない．なぜならパブリッシャーは定められたプロトコルに従いながらいつでも誤った情報を発信できるからである．よって、各サブスクライバーは購読リクエストで指定したパブリッシャーが正しいデータを提供してくれることを信頼する必要がある．

3. 安全なアグリゲーションプロトコル

本章では、段階的に我々の開発したプロトコルを説明していく．第 3.1 章では、パブリッシャーの機密データを守るプロトコルを説明する．次に第 3.2 章では合計値の正しさを保証するプロトコルを示す．最後にデータの機密性と正確性の両方の特性を保証する統合されたプロトコルを第 3.3 章に示す．

3.1 機密データを保護するアグリゲーション

各パブリッシャーの機密データを守るプロトコルを下記に示す．プロトコルを解説するにあたり、n 個のパブリッシャー $p_1, \ldots, p_n$ と単一のサブスクライバー p_{sub} が存在する場合を想定する．各パブリッシャー p_i はデータ v_i に対し、機密ポリシー $acl_i(v_i) = \emptyset$ を定義している．つまりサブスクライバー p_{sub} は p_i のデータ v_i に対するアクセス権限をもたない．しかし各パブリッシャー p_i は n 個のパブリッシャーのデータの合計値へのアクセス権限を p_{sub} に与えるとする．つまり p_{sub} はアクセスコントロールリスト $acl_i((p_1, v_1), \ldots, (p_n, v_n)) = \{p_{sub}\}$ を定義する．つまり n 個のパブリッシャーはパブサブシステムを使ってそれぞれが発信するデータの合計値をサブスクライバー p_{sub} に配信する．この場合、我々が想定する攻撃者は、最大 m 個のルーティングノードとサブスクライバー p_{sub} の組み合わせである．

各パブリッシャー p_i はデータ v_i の機密性を守るため、v_i を m 個のデータに分割し、それぞれを違うルーティングノードに送る．この際、m 個に分割されたデータの合計値はもとの v_i に等しいとする．但しこの方法では、サブスクライバー p_{sub} に一番近いルーティングノードが合計値を算出するので、合計値の機密性が守れない．そこで、各パブリッシャー p_i はある値 $q_i \in \mathbb{Z}_p$ をランダムに生成し、v_i $q_i (mod\ p)$ を m 個に分割したものを発信する．このプロトコルでは、サブスクラバー p_{sub} は各パブリッシャー p_i がどのようにして、乱数 q_i を生成するか知っており、受け取った合計値から乱数の合計値の引き算を行うことで正しい合計値を算出する．

この機密性保持のプロトコルは下記のステップから構成される．

準備段階の秘密共有． 各パブリッシャー p_i は秘密の乱数 q_i と疑似乱数生成装置を生成する．疑似乱数装置は初期値にシードから乱数列を生成する．つまり $PRNG : \mathbb{Z}_l \times \mathbb{N} \to \mathbb{Z}_p$ となる．ここで l は鍵のサイズであり、p は出力値のサイズを示す．便宜上、疑似乱数装置はタイムスタンプを２番目の引数とする．つまり $PRNG : \mathbb{Z}_l \times \mathcal{T} \to \mathbb{Z}_p$ となり、$\mathcal{T}$ は全てのタイムスタンプの集合である．

(1) 各パブリッシャーはシード q_i をランダムに生成し、q_i をサブスクライバー p_{sub} に機密性を保ちながら送信する．

データの発信． パブリッシャー p_i はデータ v_i を時間 t_l に下記の手順で発信する．

(1) $q_i(t_l) = PRNG(q_i, t_l)$ を計算する．

(2) $v'_i = v_i - q_i(t_l)$ を計算する．この手順は合計値 $\sum_{i=1}^{n} v_i$ をサブスクライバーに一番近いルーティングノードから守るために必要である．

(3) ランダムにデータ v'_i を m 個のデータ $v'_{i,1}, \ldots, v'_{i,m}$ に分割する．但し $v'_i = \sum_{j=1}^{m} v'_{i,j}$ を満足する．

(4) それぞれの分割したデータ $v'_{i,1}, \ldots, v'_{i,m}$ を m 個の異なるルーティングノードに送る．

各ルーティングノードにおける合計値の計算． 各ルーティングノードはある数のデータをパブリッシャーまたは他のルーティングノードから受け取り、それらの合計をルーティングの経路上の次のノードに渡す．

(1) 複数のデータ $v_1, \ldots, v_k$ を k 個のパブリッシャーまたはルーティングノードから受け取る．ここで k はルーティングの経路上の一つ手前のノードの数である．受け取るデータは同一のタイムスタンプ t_l をもつと仮定する．

(2) 合計値 $v = \sum_{i=1}^{k} v_i$ を計算する．

(3) 合計値とタイムスタンプのペア (v, t_l) を次のルーティングノードに送信する．

合計値の計算． セキュリティマネージャーが決めたルーティング経路では単一のルートノードが存在し、合計値 v'_{sum} を計算すると仮定する．サブスクライバー p_{sub} が v'_{sum} とタイムスタンプ t_l を受け取った際、下記の手順で正しい合計値 v_{sum} を算出する．

(1) タイムスタンプ t_l に対する乱数 $q_i(t_l)$ をそれぞれのパブリッシャー p_i に対して $PRNG(q_i, t_l)$ から算出する．

(2) 最終合計値を $v_{sum} = v'_{sum} + \sum_{i=1}^{n} q_i(t_l) = \sum_{i=1}^{n} v_i$ から求める．

通常のパブサブシステムの通信負荷は、n をパブリッシャーの数、r をルーティングノードの数とすると、全体の通信負荷は $O(n+r)$ となる．それに対し、機密性を保持する本章のプロトコルの通信負荷は $O(nm+r)$ となる．これは、各パブリッシャーが m 個のメッセージをルーティングノードに送信するためである．

定理 1 (合計値の機密性)．最大 m 個の共謀するサブスクライバー p_{sub} とルーティングノードからなる攻撃者は合計値 $\sum_{i=1}^{n} v_i$ を知ることができない．

Proof. ルーティングの経路のルートに位置するルーティングノードは合計値から各パブリッシャーが生成した乱数を引いた合計値 $v'_{sum} = \sum_{i=1}^{n} v'_i$ しか取得できない。ここで、 $v'_i = v_i - q_i(t_l)$ とする．この擬似の合計値 v'_{sum} は実際の合計値 $v_{sum} = \sum_{i=1}^{n} v_i$ と何の相関性も持たない, したがって、サブスクライバーと各パブリッシャー p_i の秘密の各乱数 q_i を知ることなく、合計値を知ることはできない． □

定理 2 (パブリッシャーのデータの機密性)．パブリッシャーはデータを m 個のデータに分割するとする．最大 m 個のルーティングノードとサブスクライバー p_{sub} から構成される攻撃者は各パブリッシャー p_i の機密のデータ v_i を知ることができない．

Proof. 一般性を損なうことなく、パブリッシャー p_i のデータ v_i について分析を行う．パブリッシャー p_i はデータ $v'_i = v_i - PRNG(q_i, t_l)$ を m 個に分割し、それぞれの分割したデータを異なるルーティングノードに送る．パブリッシャー p_i と各ルーティングノードのメッセージは両者の共有鍵で暗号化されているので、データ v'_i を復元するには、m 個のルーティングノードが攻撃者の管理下になる必要がある．

セキュリティマネージャーがルーティングの経路を作成する際、必ず、どのルーティングノードも 2 つ以上の他のノードまたはパブリッシャーからデータを受け取るように経路を決めている．従って、パブリッシャー p_i からデータを直接受け取らなかったルーティングノードが受け取るデータは常に p_i 以外のパブリッシャーのデータを足し合わされていて、p_i からの分を抽出することはできない．

攻撃者がデータ v'_i から v_i を得るためには、サブスクライバーが保持するデータ $q_i(t_l)$ が必要になり、この場合、攻撃者は $m+1$ の主体を最低必要とする．したがって、m 個以下の主体では、各パブリッシャーの機密データ v_i を知ることができない． □

3.2 正確性を保証するアグリゲーション

次に合計値の正確性を保証するアグリゲーションプロトコルを紹介する．このプロトコルでは、離散対数に基づく準同型のメッセージ認証コードを用いることでサブスクラバーが合計値の正確性を検証できるようにしている．各パブリッシャーはデータ v のメーセージ認証コードを $MAC(v, g) = g^v$ として取得する．ここで g はオーダーがある素数 p の群 G_p のジェネレータである．ここで g は各パブリッシャーとサブスクラバーの共有すう秘密であると仮定する．この仮定は、悪意のルーティングノードが合計値とメッセージ認証コードの整合性を保ちながら修正するのを防ぐためである．このメッセージ認証コードが以下のような準同型の性質をもつ．

$$MAC(v_1, g) \times MAC(v_2, g) = MAC(v_1 + v_2, g)$$

なぜなら $g^{v_1} g^{v_2} = g^{v_1+v_2}$ が成り立つからである．

合計値の正確性を保つプロトコルは下記のステップから構成される．

準備段階の秘密共有． 全てのパブリッシャーとサブスクラバー p_{sub} はセキュリティマネージャーが選んだ秘密のジェネレータ g を共有しなければならない．また各パブリッシャー p_i とサブスクラバー p_{sub} は秘密のシード r_i を共有する,

(1) セキュリティマネージャーは群 G_p のジェネレータ g をランダムに生成し、各パブリッシャー p_i とサブスクライバー p_{sub} に機密性を保ちつつ送信する．但し群 G_p のオーダー p は全ての主体の知る公の知識である．

(2) 各パブリッシャー p_i はシード r_i をランダムに生成し、機密性を保ちつつ p_{sub} に送信する．

データの発信． パブリッシャーはデータとそのメッセージ認証コードを送信する．

(1) $r_i(t_l) = PRNG(r_i, t_l)$ を計算する．

(2) メッセージ認証コード $c(v_i) = g^{v_i + r_i(t_l)}$ を計算する．

(3) $v_i, c(v_i)$ をルーティングノードに送信する．

各ルーティングノードにおける合計値の計算 各ルーティングノードはある数のデータとメッセージ認証コードのペアをパブリッシャーまたは他のルーティングノードから受け取る．ルーティングノードはデータの合計値とメッセージ認証コードの席を計算し、次のルーティングノードに渡す．各ルーティングノードでは下記の処理が行われる．

(1) データとメッセージ認証コードのペアを複数のパブリッシャーまたは他のルーティングノードから受け取る．

(2) データの合計値 $v = \sum_{i=1}^{k} v_i$ を計算する．

(3) メッセージ認証コードの積 $c = \prod_{i=1}^{k} c_i \ (mod\ p)$ を計算する．

(4) 合計値とメッセージ認証コードの積 (v, c) とタイムスタンプ t_l を次のルーティングノードに送信する．

合計値の検証． サブスクライバー p_{sub} は合計値 v_{sum} とメッセージ認証コード c_{sum} を受け取った後、合計値の正しさを下記の手順で検証する．

(1) 合計値 v_{sum} とメッセージ認証コード c_{sum} を受け取る．
(2) それぞれの乱数 $r_i(t_l)$ を $PRNG(r_i, t_l)$ から計算する．
(3) もし $g^{v_{sum}+\sum_{i=1}^{n} r_i(t_l)} \ (mod\ p) = c_{sum}$ が満たされれば、合計値 v_{sum} を受け入れる．

本章のデータの正確性を保証するプロトコルはメッセージ数を増加させない．但し、各メッセージはメッセージ認証コードが付与されるので各メッセージサイズは増加する．次にこのプロトコルが合計値の正確性を保証することを証明する．

定理 3 (合計値の正確性)．サブスクライバー p_{sub} が正しくない合計値 $v_{sum} \neq \sum_{i=1}^{n} v_i$ を受け入れる確率は、$\frac{1}{p}$ 以下である．ここで p は群 G_p の素数オーダーとする．

Proof. ここでは、全てのルーティングノードを管理下におく攻撃者を想定する．この攻撃者は、各パブリッシャーから $v_i, c_i = g^{v_i+r_i(t)}$ のペアを入手する．ここで、同じ群 G_p に属する別のジェネレータ $\hat{g} \neq g$ を考慮する．この場合、乱数 $\hat{r_i}(t)$ が存在し、 $g^{v_i+r_i(t)} = \hat{g}^{v_i+\hat{r_i}(t)}$ を満足する．さらに、擬似乱数生成装置が十分な暗号的強度をもつと想定すると、攻撃者はジェネレータ g または $\hat{g}$ のどちらが実際に使用されているか識別不能である．つまり下記の等式が成り立つ．

$$c_{sum} = g^{v_{sum}+\sum_{i=1}^{n} r_i(t)} = \hat{g}^{v_{sum}+\sum_{i=1}^{n} \hat{r_i}(t)}$$

しかし、もし $v'_{sum} \neq v_{sum}$ であれば、下記の不等式が成り立つ．

$$g^{v'_{sum}+\sum_{i=1}^{n} r_i(t)} \neq \hat{g}^{v'_{sum}+\sum_{i=1}^{n} \hat{r_i}(t)}$$

もしそうでないとすると、

$$g^{v'_{sum}+\sum_{i=1}^{n} r_i(t)}/c_{sum} = g^{v'_{sum}-v_{sum}}$$
$$= g'^{v'_{sum}+\sum_{i=1}^{n} \hat{r_i}(t)}/c_{sum} = \hat{g}^{v'_{sum}-v_{sum}},$$

となり、2つのジェネレータが異なる、つまり $g \neq \hat{g}$、という仮定に相反するからである．したがって、たとえ攻撃者が合計時 v_{sum} とジェネレータ $\hat{g}$ を適当に選んで同じメッセージ認証コード c_{sum} を計算したとしても、それが群 G_p に属する v_{sum} 以外のデータ v'_{sum} に対して正しいメッセージ認証コードである確率は、$\frac{1}{p}$ である． □

3.3 安全なアグリゲーション

最後に前章2つのプロトコルを統合した安全なアグリケーションプロトコルを説明する．このプロトコルは下記のステップから構成される．

Initial secret sharing. 各パブリッシャー p_i は2つのシード q_i and r_i をサブスクラバーと共有する．

データの発信. 各パブリッシャー p_i は値 $v_i - PRNG(q_i, t_l)$ を m 個に分割し、またメッセージ認証コードを m に分割し、その積が $MAC(v_i + PRNG(r_i, t_l), g)$ に等しくなるようにする．各パブリッシャーは下記の手順を取る．

(1) $q_i(t_l) = PRNG(q_i, t_l)$ and $r_i(t_l) = PRNG(r_i, t_l)$ を計算する．
(2) $v'_i = v_i - q_i(t_l)$ を計算する．
(3) データ v'_i を m 個のデータ $v'_{i,1}, \ldots, v'_{i,m}$ にランダムに分割し、$v'_i = \sum_{j=1}^{m} v'_{i,j}$ を満たすものとする．
(4) $v''_i = v_i + r_i(t_l)$ を計算する．
(5) データ v''_i を m 個のデータ $v''_{i,1}, \ldots, v''_{i,m}$ にランダムに分割し、$v''_i = \sum_{j=1}^{m} v''_{i,j}$ を満たすものとする．
(6) メッセージ認証コード $c(v''_{i,j}) = MAC(v''_{i,j}, g)$ を $j = 1$ から m について計算する．
(7) データとメッセージ認証コードのペア $(v'_{i,1}, c(v''_{i,1})), \ldots, (v'_{i,m}, c(v''_{i,m}))$ を m 個の異なるルーティングノードに送信する．

各ルーティングノードにおける合計値の計算. 各ルーティングノードでは下記の手順を行う．

(1) 複数のデータとメッセージ認証コードのペア $(v_1, c_1), \ldots, (v_k, c_k)$ をパブリッシャーまたは他のルーティングノードから受け取る．
(2) 合計値 $v = \sum_{i=1}^{k} v_i$ を計算する．
(3) メッセージ認証コードの積 $c = \prod_{i=1}^{k} c_i \ (mod\ p)$ を計算する．
(4) ペア (v, c) とタイムスタンプ t_l を次のルーティングノードに送信する．

合計値の計算と検証. サブスクライバーは下記の手順で合計値を求め、正確性を検証する．

(1) 各パブリッシャー p_i との2つの秘密、$r_i(t_l) = PRNG(r_i, t_l)$ と $q_i(t_l) = PRNG(q_i, t_l)$ を計算する．
(2) 合計値 $v_{sum} = v'_{sum} + \sum_{i=1}^{n} q_i(t_l) = \sum_{i=1}^{n} v_i$ を計算する．
(3) もし $g^{v_{sum}+\sum_{i=1}^{n} r_i(t_l)} = c_{sum}$ が成立すれば、合計値 v_{sum} を受け入れる．

定理 4 (データの機密性と正確性)．本章のプロトコルは各パブリッシャーのデータの機密性を守り、パブリッシャーに対し合計値の正確性を保証する．

データの機密性の証明は、証明 2 と同様であり、合計値の正確性の証明は、証明 3 と同様である．よって本定理の証明は省略する．

4. 関連する過去の研究

過去に多くの研究者がパブサブシステムのセキュリティの問題に取り組んできた．

Wang ら[14] はパブサブシステムのセキュリティの問題を概説した．幾つかの研究では、パブサブシステム上でセキュリティのポリシーを記述するためのポリシー言語の研究がなされた[6),7),15)]．より最近の研究では、パブサブシステムのルーティングノードが信頼できないと仮定して、流通する情報の機密性を守るための研究がなされた． Khurana's scheme[4] や Pesonen ら[8] は準同型の暗号を用いてデータの機密性を保持している．EventGuard[11] ではデータの機密性に加え、サブスクライバーの購読リクエストの機密性を守る包括的な仕組みを提案している．しかしながら過去のパブサブシステムでは、本研究のようにルーティングノードでアグリゲーションを行う場合が想定しておらず、情報の機密性と正確性を同時に保証するプロトコルはこれまで検討されていない．

5. おわりに

今回は、パブサブシステムのルーティングノードがデータの合計値を算出する場合に関して、パブリッシャーのデータの機密性とサブスクライバーの受け取る合計値の正確性の両方を保証するプロトコルを開発した．今後はより一般的な統計関するに対して、同様の安全性を保証するプロトコルを開発していきたい．

謝辞 この論文は、総務省 PREDICT プロジェクトと科研費基盤 C (11013869) の助成を受けたものである．

参考文献

1) Bakken, D.E., Hauser, C.H., Gjermundrod, H. and Bose, A.: Towards More Flexible and Robust Data Delivery for Monitoring and Control of the Electric Power Grid, Technical Report TR-GS-009, Washington State University (2007).
2) Carzaniga, A., Rosenblum, D.S. and Wolf, A.L.: Design and evaluation of a wide-area event notification service, *ACM Transactions on Computer Systems*, Vol.19, No.3, pp.332–383 (2001).
3) Kastner, W., Neugschwandtner, G., Soucek, S. and Newmann, M.H.: Communication Systems for Building Automation and Control, *Proceedings of the IEEE*, Vol.93, No.6, pp.1178–1203 (2005).
4) Khurana, H.: Scalable security and accounting services for content-based publish/subscribe systems, *Proceedings of the 2005 ACM symposium on Applied computing*, New York, NY, USA, ACM Press, pp.801–807 (2005).
5) McDaniel, P. and McLaughlin, S.: Security and Privacy Challenges in the Smart Grid, *IEEE Security and Privacy*, Vol.7, pp.75–77 (2009).
6) Miklos, Z.: Towards an Access Control Mechanism for Wide-Area Publish/Subscribe Systems, *Proceedings of the 22nd International Conference on Distributed Computing Systems*, Washington, DC, USA, IEEE Computer Society, pp.516–524 (2002).
7) Pesonen, L. I.W., Eyers, D.M. and Bacon, J.: A Capability-Based Access Control Architecture for Multi-Domain Publish/Subscribe Systems, *Proceedings of the International Symposium on Applications on Internet*, Washington, DC, USA, IEEE Computer Society, pp.222–228 (2006).
8) Pesonen, L. I.W., Eyers, D.M. and Bacon, J.: Encryption-enforced access control in dynamic multi-domain publish/subscribe networks, *Proceedings of the 2007 inaugural international conference on Distributed event-based systems*, New York, NY, USA, ACM, pp.104–115 (2007).
9) Ramasubramanian, V., Peterson, R. and Sirer, E.G.: Corona: A High Performance Publish-Subscribe System for the World Wide Web, *Proceedings of the 3rd Symposium on Networked Systems Design and Implementation* (2006).
10) Robert O.Burnett, J., Butts, M.M. and Sterlina, P.S.: Power system applications for phasor measurement units, *Computer Applications in Power, IEEE*, Vol.7, No.1, pp.8–13 (1994).
11) Srivatsa, M. and Liu, L.: Secure Event Dissemination in Publish-Subscribe Networks, *Proceedings of the 27th International Conference on Distributed Computing Systems*, Washington, DC, USA, IEEE Computer Society, p.22 (2007).
12) Strom, R., Banavar, G., Chandra, T., Kaplan, M., Miller, K., Mukherjee, B., Sturman, D. and Ward, M.: Gryphon: An information flow based approach to message brokering, *International Symposium on Software Reliability Engineering (ISSRE '98)* (1998).
13) Tomsovic, K., Bakken, D.E., Venkatasubramanian, V. and Bose, A.: Designing the Next Generation of Real-Time Control, Communication, and Computations for Large Power Systems, *Proceedings OF THE IEEE*, Vol.93, No.5, pp.965–979 (2005).
14) Wang, C., Carzaniga, A., Evans, D. and Wolf, A.L.: Security Issues and Requirements for Internet-Scale Publish-subscribe Systems, *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii (2002).
15) Zhao, Y. and Sturman, D.C.: Dynamic Access Control in a Content-based Publish/Subscribe System with Delivery Guarantees, *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, Washington, DC, USA, IEEE Computer Society, p.60 (2006).